# Diamond-Kite Meshes: Adaptive Quadrilateral Meshing and Orthogonal Circle Packing

David Eppstein

Department of Computer Science, University of California, Irvine
eppstein@ics.uci.edu

**Summary.** We describe a family of quadrilateral meshes based on *diamonds*, rhombi with 60° and 120° angles, and *kites* with 60°, 90°, and 120° angles, that can be adapted to a local size function by local subdivision operations. The vertices of our meshes form the centers of the circles in a pair of dual circle packings in which each tangency between two circles is crossed orthogonally by a tangency between two dual circles.

## 1 Introduction

A famous and deep theorem of Koebe, Andreev, and Thurston [1, 2, 23, 40] asserts that the vertices of every planar graph may be represented by a *circle packing*, a system of circles with disjoint interiors, such that two vertices are adjacent in the graph if and only if the corresponding two circles are tangent. This representation is not unique without additional constraints (for instance, a 4-cycle has infinitely many distinct representations as a set of four tangent circles) but it can be made unique, up to Möbius transformations, in one of two different ways:

- Let $G$ be constrained to be a *maximal* planar graph; that is, every face of $G$, including the outer face, must be a triangle. Then its representation as a circle packing exists and is unique up to Möbius transformations (Thurston, Corollary 13.6.2). We call this a *maximal circle packing*. An example is shown in Figure 1, left.
- Alternatively, let $G$ be a 3-*vertex-connected* planar graph. It has a unique planar embedding; let $G'$ be the dual graph of this embedding. Then it is possible to represent both $G$ and $G'$ by simultaneous circle packings with the property that, for every edge $e$ of $G$ and its corresponding dual edge $e'$, the two circles representing the endpoints of $e$ have the same point of tangency as the two circles representing the endpoints of $e'$ and, moreover, the circles for $e$ cross the circles for $e'$ at right angles at this point. Again,

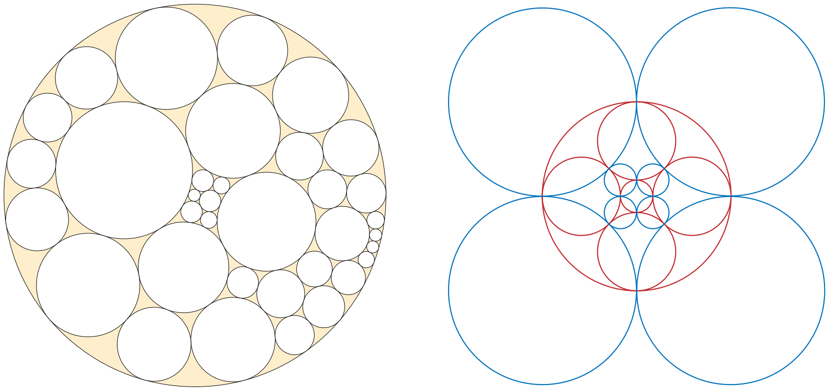

**Fig. 1.** A maximal circle packing (left) and an orthogonal circle packing (right).

this representation is unique up to Möbius transformations [10], and we call it an *orthogonal circle packing*. An example is shown in Figure 1, right.

Circle packings of these types have been applied in the field of graph drawing, to find drawings of planar graphs with right angle crossings [10], high angular resolution [13, 29], and small numbers of distinct slopes [21]. They have also been used for mesh partitioning [17,32,33], for visualization of brain structures [20], for analyzing the structure of soap bubbles [16], for solving differential equations [19], for constructing Riemann surfaces from combinatorial data [9], and for finding approximations to conformal mappings between different simply connected domains, which can be used as an important step in structured mesh generation [34, 37].

The two constrained forms of circle packing guaranteed to exist by the circle packing theorem would seem to be also a natural fit for unstructured mesh generation: in a maximal circle packing, the graph of adjacencies between tangent circles (with its vertices placed at the triangle centers) forms an unstructured triangle mesh, and in an orthogonal circle packing, the graph of adjacencies between orthogonal circles forms an unstructured quadrilateral mesh. Additionally, if the degree of a graph is bounded, then the circle packings generated from it are naturally graded in size: adjacent circles have radii whose ratio is bounded, and the triangular or quadrilateral elements derived from the packing have bounded aspect ratio. However, despite their obvious appeal, these types of circle packing have not been used in mesh generation, because the geometry of a circle packing is difficult to control: circle packings are generated from combinatorial data (a graph) rather than from geometric data (the shape of a domain to be meshed) and in general, a small localized change to the graph from which the circle packing is generated can lead to large and non-localized changes to the packing.

Circle packings are still used in many unstructured mesh generation algorithms, but they are not the types of packings described by the Koebe–

Andreev–Thurston theorem. Rather, methods for unstructured mesh generation based on circle packings have typically relied on methods that generate packings with less structure, by placing circles one at a time using a greedy algorithm [4, 7, 15, 24, 25, 27, 41], physical simulation [36], or decimation of quadtree-based oversampling schemes [30,31]. Circle packing mesh generation algorithms have been used to find nonobtuse triangular meshes for polygonal domains [7, 15] as well as bounded-aspect-ratio triangular meshes [25] and quadrilateral meshes in which all elements belong to certain special types of quadrilateral [4]. The circle packings generated by these methods can be made to have radii controlled by a local size function [25, 41], and mesh generation techniques based on these methods can be applied in higher dimensions as well [24, 28, 30, 31]. However, these circle packings are neither maximal nor orthogonal; rather, as well as having the three-sided gaps between circles that a maximal circle packing would have, they also include irregular gaps with four or more sides. Instead of generating a mesh directly from the intersection pattern of the given circles, these methods need additional vertices at points such as the circumcenters of the gaps, and they typically also need additional case analysis to handle the different possible shapes of their gaps.

In this paper, we show for the first time that it is possible to construct orthogonal circle packings, one of the two types of circle packing guaranteed to exist by the Koebe–Andreev–Thurston theorem, with geometric rather than graph-theoretic control of the position and size of the circles. Our circle packings are based on a quadrilateral mesh in which every quadrilateral is either a *diamond*, a rhombus with 60° and 120° angles, or a *kite* with 60°, 90°, and 120° angles; the circles in the packing are centered at the vertices of this mesh and have their points of tangency at the points within each quadrilateral where its two diagonals cross.

In some ways, our construction resembles more standard meshing techniques based on *quadtrees* (squares recursively subdivided into four smaller squares) [5,14,18,42], or the recursive subdivision of triangles into four smaller triangles used by the Sloan digital sky survey [39]. Like quadtrees and recursive triangle meshes, any two quadrilateral meshes formed by our construction can be transformed into each other by a sequence of simple local operations, and these local operations give the set of meshes formed in this way the structure of a distributive lattice, in which any two meshes have a coarsest common refinement and a finest common coarsening. Again, like quadtrees, these subdivision operations allow our mesh to be adapted to a *size function* specifying the element size at each point of a domain, and meshes adapted to a size function in this way have a number of elements that is within a constant factor of optimal for any bounded-aspect-ratio mesh for that size function. Like quadtrees, our meshes can as well use the same subdivision operations to adapt dynamically to varying size functions at different time steps of a finite element simulation. However, the squares of a balanced quadtree may share a boundary edge with up to eight smaller squares, and therefore require additional subdivision to produce a triangle or quadrilateral mesh; in contrast,

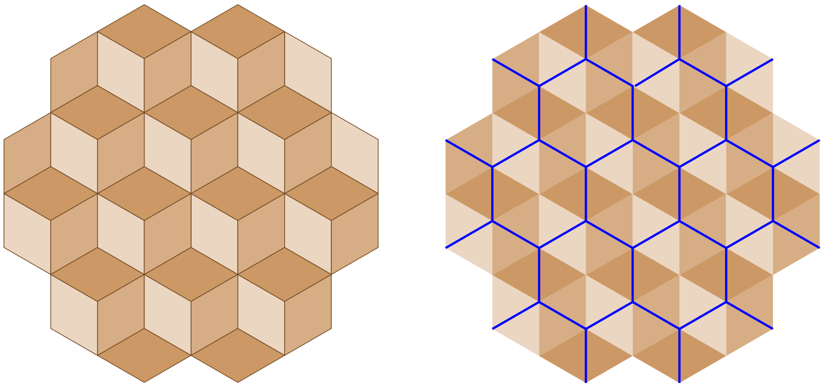

**Fig. 2.** Left: The rhombille tiling. Right: The short diagonals of the rhombille tiling form a hexagonal tiling.

our method provides a mesh directly without the need for additional subdivision. The hexagon-based meshing algorithms of Sußner, Greiner, Liang, and Zhang [26, 38] also resemble the method here, but are based on scaling by factors of two at each level of subdivision whereas we use factors of $\sqrt{3}$. Other subdivision schemes related to the methods described here include honeycomb refinement [12, 43] and $\sqrt{3}$ refinement [22, 43]; however, these schemes are typically applied to every element of a structured or semistructured mesh rather than (as here) to selected elements of an unstructured mesh.

We caution that our results should be viewed as primarily mathematical rather than being ready for immediate use in mesh generation practice. The meshes described here do not conform to domain boundaries with arbitrary slopes, and we have not implemented our algorithms.

## 2 The rhombille tiling and its local subdivision

Our construction begins with the *rhombille tiling* of Figure 2 (left), a tessellation of the plane by rhombi with 60° and 120° angles that resembles the axonometric projection of a pile of cubes and that can be formed by subdividing a regular tiling by hexagons into three rhombi per hexagon [11]. Each vertex of the rhombille tiling has degree (valency) either three or six: either six rhombi meet at their acute corners, or three rhombi meet at their obtuse corners. The short diagonals of the rhombi form another hexagonal tiling in which each hexagon surrounds a degree-six vertex of the rhombille tiling.

Suppose that we wish to refine a rhombille tiling within a region $R$ of the plane, forming a mesh with smaller elements, while leaving the tiling unchanged outside $R$. To do so, we may approximate $R$ by a set of the hexagons formed by short diagonals, and perform the local replacement operation illustrated in Figure 3 within each hexagon. This operation replaces the six

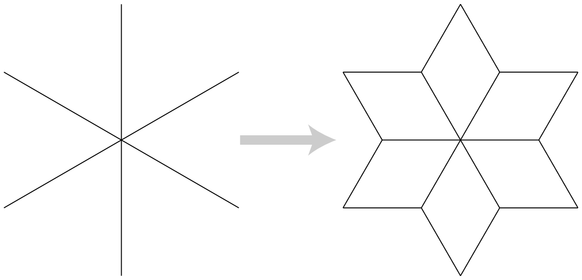

**Fig. 3.** Replacement of six rhombille-tiling edges within a hexagon of short diagonals (left) by six rhombi with side length $1/\sqrt{3}$ times the lengths of the sides in the original tiling.

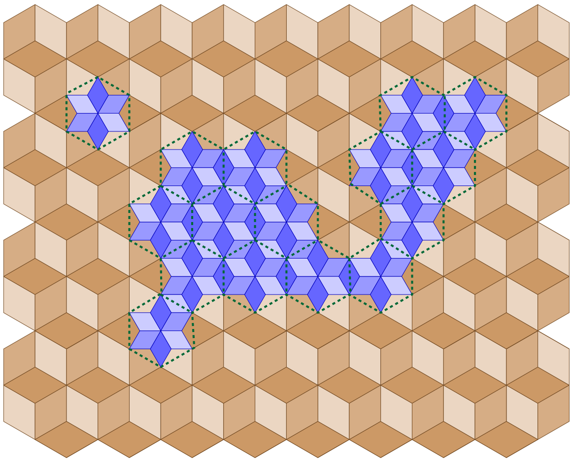

**Fig. 4.** The result of performing multiple local replacements in a rhombille tiling. Within the blue replaced region, we have another rhombille tiling, rotated from the original and with smaller tiles.

edges that meet in the center of the hexagon with a network of 18 edges, shorter by a factor of $1/\sqrt{3}$ from the original edges. These new edges form the boundaries of six rhombi similar to the ones in the original rhombille tiling but rotated from them by 30° angles. Each subdivided quadrilateral crossing the boundary of the hexagon remains a quadrilateral after the replacement, so the result of the replacement is a valid quadrilateral mesh with six additional quadrilaterals for every replaced hexagon.

Figure 4 shows the mesh resulting from the performance of multiple local replacements in a rhombille tiling. When one of the replaced hexagons abuts a hexagon that has not been replaced, the quadrilaterals formed across the shared boundary of the two hexagons are kite-shaped, with vertex angles 60°, 90°, and 120°. However, when two or more replaced hexagons abut each other,

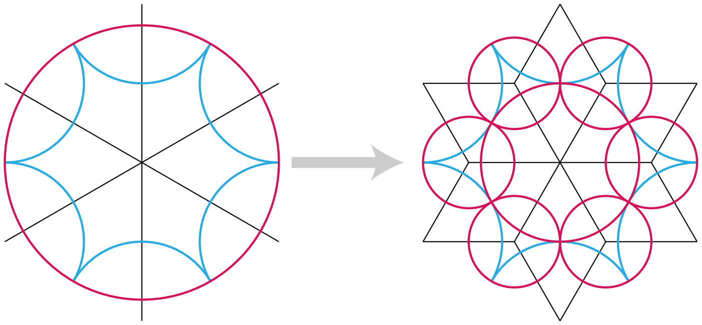

**Fig. 5.** Changes to the system of circular arcs within each quadrilateral caused by a single local replacement step.

the quadrilaterals that lie across their shared boundary are rhombi congruent to the ones contained within each replaced hexagon. Within any region formed by multiple replaced hexagons, the smaller rhombi formed by this replacement process meet up with each other in the pattern of another rhombille tiling, rotated from the original tiling by a 30° angle and with tiles that are smaller by a $1/\sqrt{3}$ factor.

Once this replacement has been performed, the same replacement process can be performed within the finer rhombille tiling formed within the replaced region. The degree-six vertices that can be replaced within this finer tiling are either the same degree-six vertices that were replaced previously, or the points where three replaced hexagons meet.

We call the meshes generated by any number of steps of this replacement process "diamond-kite meshes".

## 3 Orthogonal circle packing

We now show that the diamond-kite meshes correspond to orthogonal circle packings. In each quadrilateral of a diamond-kite mesh, place arcs of four circles, centered at the quadrilateral's four vertices and meeting at the center of the quadrilateral. Then, the circular arcs for the quadrilaterals meeting at a vertex will necessarily link up to form a single circle. In the unsubdivided rhombille tiling, this is true because the quadrilaterals sharing a vertex are all rotated images of each other, and it remains true in each of the local replacement steps by which the subdivided tiling is formed. As shown in Figure 5, the circular arcs surrounding each vertex of the replaced hexagon (shown as green in the figure) retain their previous radius, and the circular arcs surrounding the center vertex and each newly added vertex (shown as violet) meet up to form a circle that lies entirely within the replacement region.

The circles formed in this way meet in tangent pairs at the points within each quadrilateral where the diagonals cross, and (because the rhombs and

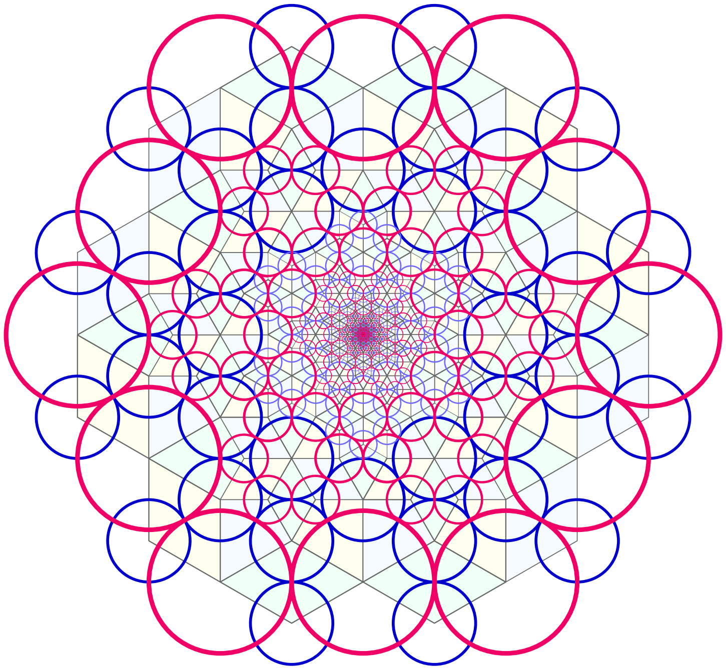

**Fig. 6.** A diamond-kite mesh with quadrilaterals of many different scales and its circle packing.

kites of the mesh are both orthodiagonal) the two pairs of tangent circles meeting at each crossing point are orthogonal to each other. Thus, the result is an orthogonal circle packing. Figure 6 shows a larger example.

## 4 Lattice of refinements

Given an initial rhombille tiling $T$, we may uniquely specify each of the local replacement steps used to form a diamond-kite mesh by a pair of parameters $(p, s)$, where $p$ is the center point of the replaced hexagon and $s$ is its side length. We may analyze cases to determine the conditions under which a replacement with parameters $(p, s)$ is possible:

- If $s$ is the length of the sides of the original tiles of $T$, then replacement $(p, s)$ may always be performed in every diamond-kite mesh formed from $T$ in which it has not already been performed.
- If $s$ is smaller than the side length in $T$, and $(p, s\sqrt{3})$ is the pair of parameters for another replacement step, then replacement $(p, s)$ may be performed if and only if replacement $(p, s\sqrt{3})$ has already been performed. The reason for this is because the replacement $(p, s\sqrt{3})$ is the only possible

way to incorporate into the tiling the six edges that will be replaced by $(p, s)$.

- In the remaining case, there are three points $p_0$, $p_1$, and $p_2$ equally spaced around $p$ at distance $s\sqrt{3}$ from it, such that each point $p_i$ gives the parameterization of a replacement step $(p_i, s\sqrt{3})$ and such that replacement $(p, s)$ may be performed if and only if all three of $(p_i, s\sqrt{3})$ have already been performed. Each of the three replacements $(p_i, s\sqrt{3})$ is the only possible way of incorporating into the tiling two of the six edges that will be replaced by $(p, s)$.

We may encode this case analysis by an infinite directed acyclic graph $G_T$ in which we have a vertex for each pair $(p, s)$ that defines a valid replacement step, and in which we have an edge from $(p, s)$ to $(p, s\sqrt{3})$ (when $(p, s\sqrt{3})$ defines a valid replacement step) or from $(p, s)$ to each of the three vertices $(p_i, s\sqrt{3})$ (in the cases where these three vertices are defined). In $G_T$, the replacement steps corresponding to all incoming neighbors of a vertex must be performed before the replacement step corresponding to the vertex itself may be performed.

Alternatively and equivalently, we may use an infinite partially ordered set $P_T$ that has an element for each pair $(p, s)$, and in which two elements $x$ and $y$ are ordered $x \leq y$ whenever there is a directed path from $x$ to $y$ in $G_T$. This formulation is more convenient for describing the set $\{x \mid x < y\}$ of all replacement steps that must be performed prior to performing operation $y$, either because they directly precede $y$ (as in the case analysis above) or because they precede one of the predecessors of $y$.

If $T'$ is a diamond-kite mesh formed by refining the initial mesh $T$, then (by the analysis above) the set of replacement steps that were used to generate $T'$ from $T$ must be a finite *lower set* in $P_T$, that is, a set $L$ of elements with the property that, for every element $y \in L$ and for every element $x$ with $x \leq y$, $x$ is also in $L$. Conversely, every finite lower set $L$ uniquely defines a diamond-kite mesh $T_L$ as a refinement of $T$, because the replacement operations in $L$ may be performed in any order that is consistent with the case analysis above; different orderings of the same operations will always lead to the same results. The orderings in which a given lower set $L$ of replacement steps may be performed are exactly the linear extensions of the partial order induced by the elements of $L$ in $P_T$, and a valid ordering for a given lower set $L$ may be calculated by applying a topological sorting algorithm to the directed acyclic graph induced as a subgraph of $G_T$ by the elements of $L$.

Let $T_1$ and $T_2$ be any two diamond-kite meshes formed by refining $T$, and described by the respective finite lower sets $L_1$ and $L_2$ in $P_T$. Then the sets $L_1 \cap L_2$ and $L_1 \cup L_2$ are also finite lower sets, describing respectively the finest mesh $T_1 \wedge T_2$ from which both $T_1$ and $T_2$ can be formed by refinement, and the coarsest mesh $T_1 \vee T_2$ that can be formed by refining both $T_1$ and $T_2$. In this way, as with the lower sets of every partially ordered set [8], the family of diamond-kite meshes can be given the structure of a distributive lattice.

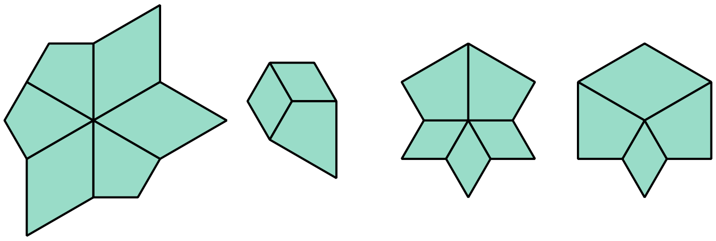

**Fig. 7.** Cases for refine($p$). From left to right: (a) $p$ has degree six, and is ready for immediate replacement; (b) $p$ has degree three, and does not meet the preconditions of refine; (c) $p$ has degree five, and a replacement at the 60° kite vertex must be performed prior to a replacement at $p$; and (d) $p$ has degree four, and two 60° kite vertices must be replaced prior to a replacement at $p$.

We remark that the same ideas of constructing an infinite graph describing the prerequisite relation between potential replacement steps, deriving an infinite partial order from the graph, and describing each possible mesh as a lower set of this partial order, can be applied equally well to describe the set of possible balanced quadtrees derived from an initial square. Thus, the set of balanced quadtrees can also be given the structure of a distributive lattice.

## 5 Local replacement at mesh vertices

When we adapt a mesh to a local size function, it will be convenient to specify the local replacement steps of the adaption process by vertices of a mesh rather than by the more abstract elements of an infinite partially ordered set described in the previous section.

We will define a recursive subdivision algorithm refine($p$) that performs a single local replacement step at a vertex $p$ of a diamond-kite mesh, after performing all prerequisite replacements, as follows. We require, as a precondition for this algorithm, that $p$ be a 60° vertex of at least one mesh element. Based on this requirement, the case analysis below describes which other replacement steps are necessary before the one at $p$ can be performed:

- If $p$ has degree six, then the most recent replacement step that affected the edges incident to $p$ must either have replaced a hexagon with $p$ at its center, or have been the third of three replacements of hexagons meeting at $p$. In this case, $p$ is already surrounded by diamonds and/or kites having 60° angles at $p$, as shown in Figure 7(a). It is possible to perform a local replacement step at $p$ without performing any other replacements.
- If $o$ has degree three, then it must be the case that the most recent replacement step at $p$ replaced a hexagon for which $p$ was interior but not central. Then $p$ is a 120° vertex of three elements, either two diamonds and one larger kite (as in Figure 7(b)) or three diamonds. In this case,

it is not possible for the precondition of the refine algorithm to be met, because there is no 60° angle at $p$.

- If $p$ has degree five, then it must be the case that the most recent replacement to affect the neighborhood of $p$ was the replacement of a hexagon having $p$ as a vertex, and additionally this must have been the second replacement of the three hexagons of that size meeting at $p$. Then the neighborhood of $p$ consists of three elements with 60° angles (diamonds or kites within the two replaced hexagons) and two elements with 90° angles (necessarily kites in the third hexagon); see Figure 7(c). Let $q$ be the shared 60° vertex of these two kites. We recursively call refine($q$), after which we may perform a replacement step at $p$.
- If $p$ has degree four, then $p$ was a vertex of the hexagon for the most recent replacement at $p$, and this was the first replacement of the three hexagons of that size meeting at $p$. In this case $p$ has a neighorhood with one 60° angle (a diamond or kite in the replaced hexagon), two 90° angles (kites in the two unreplaced hexagons), and one 120° angle (a rhomb or kite overlapping the two unreplaced hexagons), as is depicted in Figure 7(d). We recursively call refine for each of the two 60° vertices of the kites with incident 90° angles, after which we may perform a replacement step at $p$.

Thus, refine($p$) need only list all kites that have 90° angles at $p$, recursively call itself on the 60° angles of these kites, and then perform a replacement step at $p$. The recursion necessarily terminates, because each recursive call leads to a replacement step on a larger hexagon. Each recursive call adds elements to the mesh for the replacement step it performs, so the total time for this recursive procedure is linear in the total change to the number of elements in the mesh. The case analysis above shows that this recursion performs exactly the replacement steps that are predecessors of $(p, s)$ in the partial order $P_S$ and that had not already been performed at the start of the recursion.

## 6 Adaption to a local size function

We define a *local size function* to be a function $\sigma$ that maps each point $p$ of the plane (or of a subset of the plane to be meshed) to a positive real number $\sigma(p)$, specifying the largest allowable side length of a mesh element containing $p$.[1] We assume that access to $\sigma$ is via a subroutine oversized($Q$) that takes as argument a quadrilateral $Q$ and returns a Boolean value, true if $Q$ contains a point $p$ for which $\sigma(p)$ is less than the side length of $Q$ and false otherwise. Our task is to find a mesh that is as coarse as possible subject to the constraint that oversized returns false for all mesh elements.

[1] It would be equivalent to within constant factors to specify the maximum allowable area, perimeter, diameter, or circumradius of the element, but side length turns out to be more convenient for our purposes, because it leads to fewer ambiguities about which replacement steps are necessary.

To do so, from an initial mesh, we make a queue of unprocessed quadrilaterals in the mesh. We repeatedly find and remove a quadrilateral $Q$ from this queue, process it, and add to the queue all new quadrilaterals formed during processing, until the queue becomes empty. To process a quadrilateral $Q$, we first try calling oversized($Q$); if it returns false, processing is complete. Otherwise, if $Q$ is a kite, we let $p$ be its 60° vertex and call refine($p$). If $Q$ is a diamond, let $p_1$ and $p_2$ be its 60° vertices and let $Q_1$ and $Q_2$ be the kites within $Q$ that have the same maximum side length as $Q$ and that have $p_1$ and $p_2$ (respectively) as their 60° vertices. For each $i$ in the set $\{1,2\}$ we call oversized($Q_i$) and, if it returns true, we call refine($p_i$). Note that it is not possible for both oversized($Q_1$) and oversized($Q_2$) to be false, because that would be inconsistent with the assumed true value of oversized($Q$). Thus, regardless of the shape of $Q$, if oversized($Q$) is true then processing $Q$ will cause it to become subdivided.

Whenever this adaption procedure calls refine($p$), leading to a replacement step at vertex $p$, the same replacement step must be performed in all diamond-kite meshes that obey the size function $\sigma$, because otherwise the current quadrilateral $Q$ or a kite within it would remain and have too large a side length. Therefore, the result of the adaption procedure is the coarsest diamond-kite mesh consistent with the size function. Each step either removes a quadrilateral from the queue without adding any others, or it takes time linear in the number of elements added, so the total time for this adaption procedure is linear in the size of the final mesh it produces.

## 7 Size and length optimality

Following Ruppert [35], we may use *local feature size* to prove that the method for size adaption discussed in the previous section produces meshes that (compared to any other mesh with quadrilaterals or triangles of bounded aspect ratio obeying the constraints of the local size function) are within a constant factor of optimal with respect both to their number of elements and to their total edge length, matching known results for quadtree meshes [5, 14] and for meshes formed by Delaunay refinement [35].

We assume that the size function $\sigma(p)$ is defined in such a way as to lead to a finite mesh, and we define the *local feature size* to be a function $\hat{\sigma}(p)$ that maps a point $p$ to the number

$$\hat{\sigma}(p) = \inf\bigl\{\text{distance}(p,q) + \sigma(q)\bigr\}.$$

The point $q$ in the minimization ranges over the rest of the plane, but its minimum will necessarily occur within a disk of radius $\sigma(p)$ centered at $p$, because all other points lead to larger values than the value $\sigma(p)$ achieved at $q = p$. The following two observations are central to our analysis:

- In all meshes with bounded-aspect-ratio elements, the size of the element containing a given point $p$ is $\Omega(\hat{\sigma}(p))$. To see this, let $q$ be a point achieving (or approximately achieving) the minimum value in the definition of $\hat{\sigma}(p)$. The element containing $q$ must have size at most $\hat{\sigma}(p)$, and the sequence of elements crossed by the line segment from $q$ to $p$ cannot increase in size from that value by more than a constant factor before they reach $p$.

- In the diamond-kite mesh defined from the size function $\sigma$, the size of the element containing $p$ is $O(\hat{\sigma}(p))$. This follows from the fact that, if $q$ is any point in the plane, the size of the smallest element of the partial order $P_T$ that is forced by the value of $\sigma(q)$ to be included in the mesh and that corresponds to a local replacement for a hexagon containing $p$ is $O(\text{distance}(p, q) + \sigma(q)$.

Given a bounded-aspect-ratio mesh $M$, let $\alpha(p)$ denote the area of the element containing $p$. Then, for all elements $E$ of $M$, we have the identity

$$\int_E \frac{1}{\alpha(p)} dp = 1.$$

Therefore, the number of elements in the mesh can be counted by

$$\int_M \frac{1}{\alpha(p)} dp.$$

However, $1/\alpha(p)$ is lower-bounded by $\Omega((\hat{\sigma}(p))^{-2})$ for all bounded-aspect-ratio meshes, and upper-bounded by $O((\hat{\sigma}(p))^{-2})$ for diamond-kite meshes; therefore, the number of elements in the diamond-kite mesh determined by size function $\sigma$ is within a constant factor of optimal.

Similarly, given a bounded-aspect-ratio mesh $M$, let $\pi(p)$ denote the perimeter of the element containing $p$. Then, for all elements $E$ of $M$, we have the identity

$$\int_E \frac{\pi(p)}{\alpha(p)} dp = \text{perimeter}(p).$$

Therefore, the total perimeter of all the elements in the mesh is

$$\int_M \frac{\pi(p)}{\alpha(p)} dp.$$

However, $\pi(p)/\alpha(p)$ is lower-bounded by $\Omega(1/\hat{\sigma}(p))$ for all bounded-aspect-ratio meshes, and upper-bounded by $O(1/\hat{\sigma}(p))$ for diamond-kite meshes; therefore, the total perimeter of the elements in the diamond-kite mesh determined by size function $\sigma$ is within a constant factor of optimal.

## 8 Additional properties

Every vertex of a diamond-kite mesh has degree at most six. Every two orthogonal circles have radii differing by a factor of exactly $\sqrt{3}$, and every two tangent circles have radii differing by a factor of at most three.

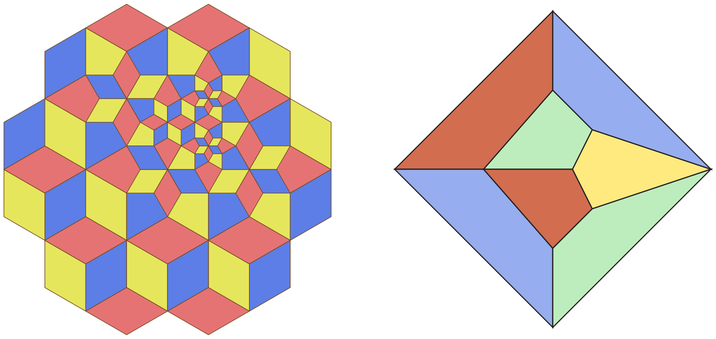

**Fig. 8.** Left: A 3-colored diamond-kite mesh. Right: A quadrilateral mesh that requires four colors in any face coloring.

Graph colorings of meshes may be used to schedule batches of parallel updates to the values stored at mesh elements, in order to ensure that each two values that are updated in the same batch are independent from each other [3,6]. As with all planar graphs in which every face has an even number of sides, the vertices of a diamond-kite mesh may be colored with two colors, but in this context it is more relevant to color the faces of the mesh so that no two faces that share an edge have the same color. This may be done with three colors by the following simple strategy: define an equivalence relation on the quadrilaterals of the mesh, according to which two quadrilaterals are equivalent when their diagonals are parallel, and assign one color to each equivalence class. There are only three equivalence classes: quadrilaterals in two different equivalence classes will have their diagonals rotated by 30° from each other, and after three such rotations we return to the starting equivalence class. No two adjacent quadrilaterals in a diamond-kite mesh may have parallel diagonals, so adjacent quadrilaterals are always assigned distinct colors. Therefore, the result is a proper 3-coloring. In contrast, quadrilateral meshes other than the diamond-kite meshes may sometimes require four colors (Figure 8).

## 9 Conclusions

We have defined the family of diamond-kite meshes based on a simple local replacement step starting with the rhombille tiling. In these meshes, the most acute angle is 60°, the most obtuse angle is 120°, and all elements have bounded aspect ratio. The element size can be controlled by a local size function, and the number of elements and total edge length of the elements is within a constant factor of optimal for the given size function. Replacement operations may be performed adaptively to handle time-dependent size functions. Unlike adaptive quadtree meshes, this system provides a quadrilateral mesh directly without any need for additional subdivision.

The most novel feature of our new meshes is that their vertices form the centers of an orthogonal circle packing of the type guaranteed to exist by the Koebe–Andreev–Thurston circle packing theorem, showing for the first time that it is possible to incorporate this type of circle packing into a meshing algorithm.

Much remains to be studied in this area. On the mathematical side, we still do not know whether it is possible to define an analogous local replacement scheme that would allow the generation of maximal circle packings (in which every gap between three circles has exactly three sides) with similar properties to those of the diamond-kite mesh, and in particular with the ability to control the size of the circles in one part of the packing without changing the geometry of the circles in distant parts of the packing. On the more practical side, it would be of interest to develop the diamond-kite method into a practical mesh generation system and to compare empirically the quality of meshes generated in this way with those from other comparable systems such as quadtrees and Delaunay refinement. We leave such developments to future research.

### Acknowledgements

This research was supported in part by the National Science Foundation under grants 0830403 and 1217322, and by the Office of Naval Research under MURI grant N00014-08-1-1015.

## References

1. E. M. Andreev. Convex polyhedra in Lobačevskiĭ spaces. *Mat. Sb. (N.S.)* 81(123):445–478, 1970.
2. E. M. Andreev. Convex polyhedra of finite volume in Lobačevskiĭ space. *Mat. Sb. (N.S.)* 83(125):256–260, 1970.
3. M. Benantar, J. E. Flaherty, and M. S. Krishnamoorthy. Coloring procedures for finite element computation on shared-memory parallel computers. *Proc. Symp. on Adaptive, Multilevel, and Hierarchical Computation Strategies.* ASME, AMD 157, 1992.
4. M. Bern and D. Eppstein. Quadrilateral meshing by circle packing. *Proc. 6th International Meshing Roundtable*, pp. 7–20. Sandia National Laboratories, 1997, arXiv:cs.CG/9908016, `http://www.imr.sandia.gov/papers/abstracts/Be49.html`.
5. M. Bern, D. Eppstein, and J. Gilbert. Provably good mesh generation. *J. Computer & Systems Sciences* 48(3):384–409, 1994, doi:10.1016/S0022-0000(05)80059-5.
6. M. Bern, D. Eppstein, and B. Hutchings. Algorithms for coloring quadtrees. *Algorithmica* 32(1):87–94, 2002, doi:10.1007/s00453-001-0054-2, arXiv:cs.CG/9907030.
7. M. Bern, S. A. Mitchell, and J. Ruppert. Linear-size nonobtuse triangulation of polygons. *Discrete & Computational Geometry* 14:411–428, 1995.

8. G. Birkhoff. Rings of sets. *Duke Mathematical Journal* 3(3):443–454, 1937, doi:10.1215/S0012-7094-37-00334-X.
9. P. L. Bowers and K. Stephenson. *Uniformizing dessins and Belyĭ maps via circle packing.* Memoirs of the American Mathematical Society 805. American Mathematical Society, 2004.
10. G. R. Brightwell and E. R. Scheinerman. Representations of planar graphs. *SIAM Journal on Discrete Mathematics* 6(2):214–229, 1993, doi:10.1137/0406017.
11. J. H. Conway, H. Burgiel, and C. Goodman-Strass. Chapter 21: Naming Archimedean and Catalan polyhedra and tilings. *The Symmetries of Things.* A.K. Peters, 2008.
12. N. Dyn, D. Levin, and D. Liu. Interpolatory convexity-preserving subdivision schemes for curves and surfaces. *Computer-Aided Design* 24(4):211–216, 1992, doi:10.1016/0010-4485(92)90057-H.
13. D. Eppstein. Planar Lombardi drawings for subcubic graphs. To appear in *Proc. Int. Symp. Graph Drawing (GD 2012)*, arXiv:1206.6142.
14. D. Eppstein. Approximating the minimum weight Steiner triangulation. *Discrete & Computational Geometry* 11(2):163–191, 1994, doi:10.1007/BF02574002.
15. D. Eppstein. Faster circle packing with application to nonobtuse triangulation. *Internat. J. Comput. Geom. Appl.* 7(5):485–491, 1997, doi:10.1142/S0218195997000296.
16. D. Eppstein. The graphs of planar soap bubbles, arXiv:1207.3761. Submitted, 2012.
17. D. Eppstein, G. L. Miller, and S.-H. Teng. A deterministic linear time algorithm for geometric separators and its applications. *Fundamenta Informaticae* 22(4):309–331, 1995, doi:10.3233/FI-1995-2241.
18. P. J. Frey and L. Maréchal. Fast adaptive quadtree mesh generation. *Proc. 7th International Meshing Roundtable*, pp. 211–224. Sandia National Laboratories, 1998, `http://www.imr.sandia.gov/papers/abstracts/Fr104.html`.
19. Z.-X. He. Solving Beltrami equations by circle packing. *Transactions of the American Mathematical Society* 322(2):657–670, 1990, doi:10.2307/2001719.
20. M. K. Hurdal, P. L. Bowers, K. Stephenson, D. W. L. Sumners, K. Rehm, K. Schaper, and D. A. Rottenberg. Quasi-conformally flat mapping the human cerebellum. *Proc. Medical Image Computing and Computer-Assisted Invervention (MICCAI '99)*, pp. 279–286. Springer, Lecture Notes in Computer Science 1679, 1999, doi:10.1007/10704282_31.
21. B. Keszegh, J. Pach, and D. Pálvölgyi. Drawing Planar Graphs of Bounded Degree with Few Slopes. *Proc. Int. Symp. Graph Drawing (GD 2010)*, pp. 293–304. Springer, Lecture Notes in Computer Science 6502, 2011, doi:10.1007/978-3-642-18469-7_27.
22. L. Kobbelt. $\sqrt{3}$-subdivision. *Proc. 27th Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, pp. 103–112, 2000, doi:10.1145/344779.344835.
23. P. Koebe. Kontaktprobleme der Konformen Abbildung. *Ber. Sächs. Akad. Wiss. Leipzig, Math.-Phys. Kl.* 88:141–164, 1936.
24. X.-Y. Li, S.-H. Teng, and A. Üngör. Biting spheres in 3D. *Proc. 8th International Meshing Roundtable*, pp. 85–95. Sandia National Laboratories, 1999, `http://www.imr.sandia.gov/papers/abstracts/Li133.html`.

25. X.-Y. Li, S.-H. Teng, and A. Üngör. Biting: advancing front meets sphere packing. *International Journal for Numerical Methods in Engineering* 49(1-2):61–81, 2000.
26. X. Liang and Y. Zhang. Hexagon-based all-quadrilateral mesh generation with guaranteed angle bounds. *Computer Methods in Applied Mechanics and Engineering* 200(23–24):2005–2020, 2011, doi:10.1016/j.cma.2011.03.002.
27. J. Liu, S. Li, and Y. Chen. A fast and practical method to pack spheres for mesh generation. *Acta Mechanica Sinica* 24(4):439–447, 2008, doi:10.1007/s10409-008-0165-y.
28. S. H. Lo and W. X. Wang. Generation of tetrahedral mesh of variable element size by sphere packing over an unbounded 3D domain. *Computer Methods in Applied Mechanics and Engineering* 194(48–49):5002–5018, 2005, doi:10.1016/j.cma.2004.11.022.
29. S. Malitz and A. Papakostas. On the angular resolution of planar graphs. *SIAM Journal on Discrete Mathematics* 7(2):172–183, 1994, doi:10.1137/S0895480193242931.
30. G. L. Miller, D. Talmor, S.-H. Teng, and N. Walkington. A Delaunay based numerical method for three dimensions: generation, formulation, and partition. *Proce. 27th ACM Symp. on Theory of Computing (STOC '95)*, pp. 683–692, 1995, doi:10.1145/225058.225286.
31. G. L. Miller, D. Talmor, S.-H. Teng, and N. Walkington. On the radius-edge condition in the control volume method. *SIAM Journal on Numerical Analysis* 36(6):1690–1708, 1999, doi:10.1137/S0036142996311854.
32. G. L. Miller, S.-H. Teng, W. P. Thurston, and S. A. Vavasis. Separators for sphere-packings and nearest neighbor graphs. *Journal of the ACM* 44(1):1–29, 1997, doi:10.1145/256292.256294.
33. G. L. Miller, S.-H. Teng, W. P. Thurston, and S. A. Vavasis. Geometric separators for finite-element meshes. *SIAM Journal on Scientific Computing* 19(2):364–386, 1998, doi:10.1137/S1064827594262613.
34. B. Rodin and D. Sullivan. The convergence of circle packings to the Riemann mapping. *Journal of Differential Geometry* 26(2):349–360, 1987, `http://projecteuclid.org/getRecord?id=euclid.jdg/1214441375`.
35. J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms* 18(3):548–585, 1995, doi:10.1006/jagm.1995.1021.
36. K. Shimada and D. C. Gossard. Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. *Proc. 3rd ACM Symp. Solid Modeling and Applications (SMA '95)*, pp. 409–419, 1995, doi:10.1145/218013.218095.
37. K. Stephenson. The approximation of conformal structures via circle packing. *Computational Methods and Function Theory 1997 (Nicosia)*, pp. 551–582. World Scientific Publishing, Ser. Approx. Decompos. 11, 1999.
38. G. Sußner and G. Greiner. Hexagonal Delaunay triangulation. *Proc. 18th International Meshing Roundtable*, pp. 519–538. Springer, 2009, doi:10.1007/978-3-642-04319-2_30.
39. A. S. Szalay, P. Z. Kunszt, A. Thakar, J. Gray, D. Slutz, and R. J. Brunner. Designing and mining multi-terabyte astronomy archives: the Sloan Digital Sky Survey. *Proc. ACM International Conf. on Management of Data (SIGMOD '00)*, pp. 451–462, 2000, doi:10.1145/342009.335439.

40. W. P. Thurston. *The Geometry and Topology of Three-Manifolds.* Mathematical Sciences Research Inst., 2002, `http://library.msri.org/books/gt3m/`. See especially Section 13.6, Andreev's theorem and generalizations.
41. W. X. Wang, C. Y. Ming, and S. H. Lo. Generation of triangular mesh with specified size by circle packing. *Advances in Engineering Software* 38(2):133–142, 2007, doi:10.1016/j.advengsoft.2006.04.006.
42. M. A. Yerry and M. S. Shephard. A modified quadtree approach to finite element mesh generation. *IEEE Computer Graphics and Applications* 3(1):39–46, 1983, doi:10.1109/MCG.1983.262997.
43. D. Zorin. Subdivision zoo. *Subdivision for Modeling and Animation*, pp. 65–102, SIGGRAPH course notes, 1999.